Package 'AdaJoint'

December 5, 2014

Title Adaptive Joint Test

Version 0.1.9

Date 2014-12-05

Author Kai Yu, Han Zhang

Description Powerful gene-based test via variable selection for genome-wide association studies

Maintainer Bill Wheeler <wheelerb@imsweb.com>

Depends mvtnorm

License GPL-2

Archs i386, x64

R topics documented:

AdaJoint *Powerful gene-based test via variable selection for genome-wide association studies*

Description

An R package for computing gene p-values using the Adaptive Joint Test. This package can be used to analyze genes and pathways based on a genetic association study with an outcome.

Details

Single-marker analysis that evaluates one genetic marker a time is the most commonly used approach for the identification of disease susceptibility loci in genome-wide association (GWA) studies. Increasing empirical evidence has suggested that there are regions or genes consisting of multiple genetic variants, which jointly contribute to the disease risk. A gene-based test, which evaluates the association between the outcome and all SNPs in the gene simultaneously, can be a more effective approach than the single-marker approach for studying such regions, thus is helpful to uncover some of missing heritability.

The package contains 3 types of gene-based tests: AdaJoint, AdaJoint2, and ARTP. The AdaJoint test starts the greedy search with the best marginal model with one SNP, hence its performance partially depends on the power of marginal test. An alternative strategy is starting from the best model with two SNPs through an exhaustive search (AdaJoint2). ARTP is the Adaptive Rank Truncated Product test.

The main function is `adajoint` which allows the user to pass in a data frame which contains all the data, or allows the data to be stored in files and have the function read in the data. If the user already has files containing test statistics for each gene, then the function `pathway.pvals` can be called to compute the gene and pathway p-values. This package also includes 4 gene databases: hg16, hg17, hg18 and hg19 which can be used with TPED genotype files.

Author(s)

Kai Yu <yuka@mail.nih.gov> and Han Zhang <han.zhang2@nih.gov>

References

Zhang H, Liang F, Wheeler W, Shi J, Yu K, Powerful gene-based test via variable selection for genome-wide association studies, submitted

Yu K, Li Q, Bergen AW, Pfeiffer RM, Rosenberg PS, Caporaso N, Kraft P, Chatterjee N Pathway analysis by adaptive combination of P-values Genet Epidemiol 33(8):700-9; 2009 Dec.

adajoint *Powerful gene-based test via variable selection for genome-wide association studies*

Description

Calculate gene p-values using the Adaptive Joint Test

Usage

```
adajoint(obj, pheno.list, gene.list, op=NULL)
```

Arguments

obj	A data frame containing all variables or a list of type snp.list.
pheno.list	A list describing the covariate and response data. See pheno.list. Note that if obj is a data frame, then pheno.list$file and pheno.list$id.var do not need to be specified.
gene.list	A list describing the SNP-gene-group data or NULL. See gene.list. If NULL, then it is assumed that all SNPs in the genotype file belong to the same gene. If obj is a data frame, then gene.list cannot be NULL. If snp.list$file is a TPED file, then gene.list can also be "hg16", "hg17", "hg18", or "hg19".
op	List of options. See details.

Details

If obj is a data frame, then all variables used in the analysis should be numeric. Otherwise if obj is a list, then the genotype data defined by snp.list and phenotype data defined by pheno.list will be merged together into a data frame with the SNPs coded as the number of copies of the minor allele. Missing genotypes for a SNP will be imputed as the mean value using the non-missing genotypes (if op$impute.missing = 1). A random seed should be set before calling pathway.normal in order to reproduce results. The randomness is due to the ranking of p-values, where ties are broken randomly. The gene p-value for a gene containing only 1 SNP is computed as the SNP p-value from a logisitic regression.

The outline of this function is as follows:
1. Read in the data if not passed in directly.
2. Remove subjects with a missing value for an adjusted covariate or case-control status.
3. Remove SNPs with high missing rate.
4. Remove SNPs with low MAF.
5. Remove SNPs with low case-control by genotype cell counts (for a binary response).
6. Remove constant SNPs.
7. Remove highly correlated SNPs within each group.
8. Remove genes with high missing rate.
9. Remove redundant SNPs from each group using a singular value decomposition.
10. Impute missing genotypes (if op$impute.missing = 1).
11. Remove subjects with at least 1 missing genotype from remaining SNPs (if op$impute.missing = 0 and op$keep.na = 0).
12. Remove genes which are subsets of other genes (if op$rm.gene.subsets = 1).
13. Compute the observed and permutation test statistics.
14. Compute the gene and pathway p-values.

SNPs in high LD:
Highly correlated SNPs within a gene can cause numerical problems, so SNP filtering needs to performed in such a case. One way to accomplish this filtering is to set op$filter.R2 to a positive number so that the cor function will be called to compute the R^2 values between each pair of SNPs and remove one SNP in each pair with R^2 > op$filter.R2. Another way to filter SNPs is to set the option gene.list$exclude.snps or gene.list$include.snps.

Options list:
Below are the names for the options list op. All names have default values if they are not specified.

- `method` 1-3: 1=AdaJoint, 2=AdaJoint2, 3=ARTP. The default is 2.
- `nperm` Number of permutations. The default is 10000.
- `out.dir` Output directory for temporary files. The default is the working directory `getwd`
- `keep.na` 0 or 1 to keep subjects with missing genotypes within each group. The default is 1.
- `impute.missing` 0 or 1 to impute missing genotypes. The default is 0.
- `use.common.subs` 0 or 1 to use the same set of subjects when processing each group. This option has no effect when `impute.missing` = 1, or when `keep.na` = 1. The default is 0.
- `filter.R2` A number between 0 and 1 to filter out SNPs that are highly correlated within each group. Set to 0 (or 1) for no filtering. The default is 0.
- `min.MAF` Threshold to remove SNPs based on their minor allele frequency. SNPs with MAF < min.MAF will be removed from the analysis. The default is 0.05.
- `max.missRate` Threshold to remove SNPs based on their missing rate. SNPs with missing rate > max.missRate will be removed from the analysis. The default is 0.2.
- `gene.max.missRate` Threshold to remove genes based on their missing rate. Genes with missing rate > gene.max.missRate will be removed from the analysis. The default is 0.1.
- `rm.gene.subsets` 0 or 1 to remove genes which are subsets of other genes. The default is 1.
- `min.count` Minimum cell count in the 2X3 table of case-control status and a SNP. SNPs with any cell count < min.count will be removed from the analysis. The default is 0.
- `snp.pvalues` 0 or 1 to compute single SNP p-values from logistic regression. These p-values will be in the `gene_snp` returned object. The default is 1.
- `print` 0 or 1 to print information to the console. The default is 1.
- `delete` 0 or 1 to delete (temporary) files containing the test statistics for each gene. The default is 1.
- `id.str` Character string that is appended to temporary file names. The default is "".

Options for gene-based tests:

- `inspect.snp.n` The number of candidate truncation points to inspect the top SNPs in a gene. The default is 2.
- `inspect.snp.percent` A value `x` between 0 and 1 such that a truncation point will be defined at every `x` percent of the top SNPs. The default is 0 so that the truncation points will be 1:`inspect.snp.n`.

Options for pathway-based test:

- `inspect.gene.n` The number of candidate truncation points to inspect the top genes in the pathway. The default is 10.
- `inspect.gene.percent` A value `x` between 0 and 1 such that a truncation point will be defined at every `x` percent of the top genes. The default is 0.05.

Assume the number of SNPs in a gene is 100. Below are examples of the truncation points for different values of `inspect.snp.n` and `inspect.snp.percent`.

inspect.snp.n	inspect.snp.percent	truncation points
1	0	1
1	0.05	5
1	0.25	25
1	1	100

2	0	1, 2
2	0.05	5, 10
2	0.25	25, 50
2	1	100
3	0.2	20, 40, 60

Value

The returned value is a list with names "gene.table", "pathway.pvalue", "most.sig.snps", "nperm", "most.sig.genes", "gene_snp" and "remove.n.subjects". Other possible names in the returned list are "remove.max.missRate", "remove.min.MAF", "remove.min.count", "remove.constant", "remove.filter.R2", and "error.groups". gene.table is a data frame containing the gene name, number of SNPs in the gene that were included in the analysis, and the p-value for the gene. gene_snp is a matrix of of the actual SNPs that were included in each gene for the analysis along with the SNP p-values (if `op$snp.pvalues` = 1). most.sig.genes are the most significant genes in the pathway. most.sig.snps is a matrix which contains the most significant SNPs for each gene with the p-values for each cut-point. If M is the value of the column Best_1toN for a given gene, then the most significant SNPs for each gene are the SNPs in columns SNP.1 - SNP.M.

References

Zhang H, Liang F, Wheeler W, Shi J, Yu K, Powerful gene-based test via variable selection for genome-wide association studies, to appear

Yu K, Li Q, Bergen AW, Pfeiffer RM, Rosenberg PS, Caporaso N, Kraft P, Chatterjee N Pathway analysis by adaptive combination of P-values Genet Epidemiol 33(8):700-9; 2009 Dec.

See Also

`pathway.pvals`

Examples

```
# Load the sample data
data(data, package="AdaJoint")

# Define pheno.list
pheno.list <- list(response.var="Y", main.vars=c("X1", "X2"))

# Define the gene-SNP list.
# A group variable need not be specified for gene level tests.
gs_file <- system.file("sampleData", "gene_data.txt", package="AdaJoint")
gene.list <- list(file=gs_file, delimiter="\t", header=1,
                  snp.var="SNP", gene.var="Gene")

# Set the options
nperm <- 100
temp.dir <- "C:/temp/"
op <- list(nperm=nperm, out.dir=temp.dir)

set.seed(123)
# Pass in a data frame to the adajoint function
#adajoint(data, pheno.list, gene.list, op=op)
```

```
# Define snp.list
geno_file <- system.file("sampleData", "geno_data.txt", package="AdaJoint")
snp.list <- list(file=geno_file, file.type=2, delimiter="\t")

# Add id.var and file to pheno.list
pheno.list$file <- system.file("sampleData", "pheno_data.txt", package="AdaJoint")
pheno.list$id.var <- "ID"

set.seed(123)
# Keep the data in files and have adajoint read in and merge the data
#adajoint(snp.list, pheno.list, gene.list, op=op)
```

`data` *Sample data*

Description

Sample data for `adajoint`

Details

data.rda is a data frame containing an id variable, response, 2 covariates and 50 SNPs on 500 subjects. The SNPs are coded as the number of copies of the minor allele.

Examples

```
# Load and print a subset
data(data, package="AdaJoint")

data[1:10, 1:10]
```

`gene.list` *List to describe the gene-SNP file*

Description

The list to describe the SNP-gene-group file for `adajoint`

Format

The format is a list:

file Text file containing at least 2 columns, where one column is for the SNPs and the other column is for the gene containing the SNP. No default.

delimiter The delimiter used in `file`.

gene.var Variable name or column number of the gene variable. The default is "Gene".

snp.var Variable name or column number of the SNP variable. The default is "SNP".

header 0 or 1 to denote if `file` contains a header of variable names.

include.genes List of genes to include in the analysis. The default is that all genes will be included.

exclude.genes List of genes to exclude in the analysis. The default is NULL.

include.snps List of SNPs to include in the analysis. The default is that all SNPs will be included.

exclude.snps List of SNPs to exclude in the analysis. The default is NULL.

group.var Variable name or column number of the group variable. This variable is only useful if the user is interested in the pathway p-value (see details). The default is `gene.var`.

Details

All the genes and SNPs listed in this file define a single pathway. SNPs in the same group are considered correlated, so the test statistics for SNPs in the same group will be generated jointly. SNPs belonging to the same gene cannot be broken up into different groups. If the genotpe data is stored in a tped file format, then `gene.list` can be one of the following: hg16, hg17, hg18, or hg19. These are gene databases from which the SNPs belonging to each gene are determined by the chromosome and location info in the tped file.

See Also

`hg16`, `hg17`, `hg18`, `hg19`

`gene_data` *SNP-Gene-Group data*

Description

Sample data file for the `file` argument of `gene.list`

Details

gene_data.txt is a tab delimited file that contains the SNPs within each gene and the genes within each group.

Examples

```
# Load and print the first 5 rows
data(gene_data, package="AdaJoint")

gene_data[1:5, ]
```

geno_data *Sample genotype data*

Description

Sample genotype data file for the `file` argument of `snp.list`

Details

geno_data.rda is a type 1 data file (see `file.type` in `snp.list`). This data contains 50 SNPs and 500 subjects, and is tab delimited. The first row of the data contains the subject ids. Starting from row 2, are the SNP ids and the genotypes for each subject. The genotypes are coded as AA, AG, GG.

Examples

```
# Load and print a substring the first 5 lines
data(geno_data, package="AdaJoint")

substring(geno_data[1:5], 1, 50)
```

hg16 *hg16 gene database*

Description

hg16 gene database

Details

hg16 is a gene database containing the chromosome, gene range, and gene name

Examples

```
# Load and print a subset
file <- system.file("sampleData", "hg16.rda", package="AdaJoint")
load(file)

hg16[1:10, ]
```

hg17 *hg17 gene database*

Description

hg17 gene database

Details

hg17 is a gene database containing the chromosome, gene range, and gene name

Examples

```
# Load and print a subset
file <- system.file("sampleData", "hg17.rda", package="AdaJoint")
load(file)

hg17[1:10, ]
```

hg18 *hg18 gene database*

Description

hg18 gene database

Details

hg18 is a gene database containing the chromosome, gene range, and gene name

Examples

```
# Load and print a subset
file <- system.file("sampleData", "hg18.rda", package="AdaJoint")
load(file)

hg18[1:10, ]
```

hg19 *hg19 gene database*

Description

hg19 gene database

Details

hg19 is a gene database containing the chromosome, gene range, and gene name

Examples

```
# Load and print a subset
file <- system.file("sampleData", "hg19.rda", package="AdaJoint")
load(file)

hg19[1:10, ]
```

pathway.pvals *Compute gene and pathway p-values*

Description

Calculate gene and pathway p-values

Usage

```
pathway.pvals(outfiles, genes, nSNP.gene, op=NULL)
```

Arguments

outfiles	Character vector of the names of the files created by adajoint
genes	Character vector of the gene names corresponding to outfiles
nSNP.gene	Numeric vector of the number of SNPs in each gene.
op	List of options. See details.

Details

The input arguments genes and nSNP.gene are only used in the resulting gene.table in the returned object. However, the option nperm needs to be correctly set. A random seed should be set before calling pathway.pvals in order to reproduce results. The randomness is due to the ranking of p-values, where ties are broken randomly.

Options list:
Below are the names for the options list op. All names have default values if they are not specified.

- nperm Number of permutations in each of the outfiles. The default is 10000.
- out.dir Output directory for temporary files. The default is the working directory getwd

- `inspect.gene.n` The number of candidate truncation points to inspect the top genes in the pathway. The default is 10.
- `inspect.gene.percent` A value `x` between 0 and 1 such that a truncation point will be defined at every `x` percent of the top genes. The default is 0.05.

Value

The returned value is a list with names "pathway.pvalue", "gene.table", and "nperm". pathway.pvalue is the ARTP p-value for the pathway. gene.table is a data frame containing the gene name, number of SNPs in the gene that were included in the analysis, and the p-value for the gene.

See Also

`adajoint`

`pheno.list` *List to describe the covariate and outcome data*

Description

The list to describe the covariate and outcome data for `adajoint`

Format

The format is a list:

file Covariate data file. This file must have variable names, one of which being a response variable (see `response.var`). No default.

id.var Name of the id variable. No default.

family "binomial" or "gaussian" for the type of response variable. No default.

response.var Name of the response variable. A binary response variable must be coded as 0 (control) and 1 (case). No default.

main.vars Character vector of variables names for variables in `file` that will be included in the model as main effects. The default is NULL.

delimiter The delimiter in `file`. The default is determined from the file.

in.miss Vector of character strings to define the missing values. This option corresponds to the option `na.strings` in `read.table`. The default is "NA".

Details

In this list, `response.var` must be specified. Depending on how the `adajoint` function is called, `file` and `id.var` may also need to be specified. **Missing data:** If any of the variables defined in `main.vars` or `response.var` contain missing values, then those subjects will be removed from the analysis.

pheno_data *Sample covariate and response data*

Description

Sample covariate and response data file for the `file` argument of `pheno.list`

Details

The file pheno_data.txt is a tab-delimited type 3 data set (see `file.type` in `pheno.list`). It contains the variables:

- `ID` The subject id
- `Y` Case-control status (0, 1)
- `X1` Continuous covariate
- `X2` Continuous covariate

Examples

```
# Load and print the first 5 rows
data(pheno_data, package="AdaJoint")

pheno_data[1:5, ]
```

snp.list *List to describe the genotype data*

Description

The list to describe the genotype data for `adajoint`

Format

The format is a list:

file File to use. No default.

file.type 2,3,7-12 (see details).

delimiter The delimiter used in `file`.

in.miss Vector of values to denote the missing values in `file`. The default is " " (2 blank spaces).

heter.codes Vector of codes used for the heterozygous genotype. If NULL, then it is assumed that the heterozygous genotype is of the form "AB", "Aa", "CT", ... etc, ie a 2-character string with different characters (case sensitive). The default is NULL.

subject.list List to describe the subject ids stored in a file. Only used for `file.type` = 9-12. The order of the subject ids is `subject.list$file` must match the order in the genotype data. See `subject.list`. The default is NULL.

id.var (Only for `file.type` = 3) The subject id variable. The default is 1.

Details

In this list, `file` must be specified. If the SNPs are already coded as the number of copies of the minor allele, then set `heter.codes` to 1 (the heterozygous genotype).

`Type 2` has data in the form:

	subject1	subject2	subject3
snp1	AA	CC	AC
snp2	GT	GT	GG

The first row must contain the subject ids. Starting from row 2, the first delimited field must contain the SNP id. The remaining delimited fields contain the genotypes. Rows are SNPs, columns are the subjects.
A type 7 file is a type 2 file compressed with gzip.

`Type 3` has data of the form:

id	snp1	snp2
subject1	AA	GT
subject2	CC	GT
subject3	AC	GG

A type 8 file is a type 3 file compressed with gzip.
`Types 9 and 10` are for imputed genotype files from IMPUTE2.
`Type 11` is a TPED file The first 4 columns of a TPED file should be the chromosome, SNP id, distance, and location. The distance in column 3 is not used, and the chromosome should be labeled as: 1-22, 23 or X, 24 or Y, 25 or XY, 26 or MT. Starting in column 5, are the genotypes for each subject. The order of these genotypes must match the order of the subjects in the phenotype file An example row of a TPED file would be:
1 rs3132489 0 2345643 A A A G A A G G 0 0 A A A G
In the row above, 0 denotes a missing allele. A type 12 file is a type 11 file compressed with gzip.

Examples

```
# Suppose the genotype data is a tab-delimited, type 2 file: c:/temp/data/geno1.txt.
#  Also assume the data has the additive coding 0, 1, 2 with NA as missing values.
# The below list is for processing the file.
snp.list <- list(file="C:/temp/data/geno1.txt", delimiter="\t", file.type=2,
                 heter.codes=1, in.miss=NA)
```

`subject.list` *List to describe the file of subject ids*

Description

The list to describe the file of subject ids for `snp.list`

Format

The format is: List of 5

file Subject id file. The file can be a single column of ids, or a delimited file of several columns with the ids as one of the columns. No default.

id.var Column number of variable name containing the subject ids. Use `id.var`=-1 if the file is a single column of ids.

file.type 3 or 8. Type 3 is a plain text file, type 8 is a file compressed with gzip. The default is 3.

delimiter The delimiter in `file`. The default is "".

header 0 or 1 if the file contains a header of variable names.

Details

This list is should only be used when the genotype data is a TPED format or the format of imputed genotype data from IMPUTE2. The order of the ids in this file must match the order of the genotypes in the genotype file.

Index